

CLAIMS

What is claimed is:

- 1 1. A method of operating a computer system, said computer
2 system including at least one processor, comprising.
3 establishing a plurality of memory units each
4 having a corresponding memory location;
5 executing a plurality of tasks running on said at
6 least one processor, said plurality of tasks being operable
7 to share data;
8 defining a plurality of lists for each memory
9 location;
10 determining the validity of said data in said
11 memory unit;
12 locking at least one of said plurality of lists
13 if said data is invalid;
14 inserting an entry corresponding to one of said
15 plurality of tasks onto said locked list;
16 unlocking said locked list; and
17 determining if data is inputted in said memory
18 location between said determining step and said unlocking
19 step.

1 2. A method for operating a computer system, said
2 computer system comprising at least one process,
3 comprising:
4 establishing a plurality of memory units each
5 having a corresponding memory location;
6 running a plurality of tasks on said processor,
7 said plurality of tasks being operable to share data;
8 defining a plurality of lists for each memory
9 location;
10 inserting an entry corresponding to one of said
11 plurality of tasks onto one of said plurality of lists if
12 said one list is unlocked; and
13 determining if another of said lists is unlocked
14 if said one list is locked.

1 3. A method for synchronizing processes in a computer
2 system, said computer system including at least one
3 processor, comprising:
4 establishing a plurality of memory units each
5 having a corresponding memory location
6 executing a plurality of tasks running on said
7 processor, said plurality of tasks being operable to share
8 data located in said memory units;
9 defining a plurality of lists for each memory
10 location;
11 locking at least one of said plurality of lists
12 if said data is not valid;
13 inserting an entry corresponding to one of said
14 plurality of tasks onto said locked list;
15 unlocking said locked list;
16 suspending said entered task until valid data is
17 found in said memory unit;
18 reading said valid data;
19 determining if other data is inputted in said
20 memory unit before said locking step and after said
21 unlocking step; and
22 reading said other data if it appears in said
23 memory unit.

1 4. The method of claim 3, wherein the locking step
2 further comprises activating selected other ones of said
3 plurality of tasks that are entered on said locked list.

1 5. The method of claim 3, wherein said plurality of lists
2 forms a linked list.

1 6. The method of claim 3, wherein said plurality of lists
2 is between four and eight.

1 7. The method of claim 3, further comprising transferring
2 the operation of said locked list when said locked list is
3 locked by another one of said plurality of tasks.

1 8. A computer system having enhanced concurrency,
2 comprising:
3 a plurality of processors;
4 a plurality of tasks running on said plurality of
5 processors;
6 a plurality of memory units each having a
7 corresponding memory location;
8 a plurality of lists corresponding to each of
9 said memory location;
10 wherein one of said plurality of tasks is
11 responsible for activating selected ones of said plurality
12 of tasks contained on the same list as said one task.

1 9. The system of claim 8, wherein said plurality of lists
2 form a linked list.

1 10. The system of claim 8, wherein said plurality of lists
2 is between four and eight.

1 11. The system of claim 8, wherein said computer system is
2 a multitasking or multiprocessing computer system.

1 12. A method of operating a computer system having at
2 least one processor, comprising:
3 determining the validity of data in a memory
4 unit;
5 locking a list corresponding to said memory unit
6 if said data is invalid;
7 inserting an entry corresponding to one of said
8 plurality of tasks onto said locked list;
9 unlocking said locked list; and
10 determining if data is inputted in said memory
11 location between said determining step and said unlocking
12 step.

1 13. A method for performing an operation within limits
2 upon a shared value stored in an actual value location
3 comprising:
4 providing a plurality of memory locations, two of
5 said memory locations being first and second reservation
6 memory locations and two of said memory locations being
7 limit memory locations containing limit values;
8 getting an operand;
9 performing the operation upon an affected
10 reservation register using the operand;
11 comparing a resulting value of the first
12 operation upon the affected reservation register to the
13 limit values in the limit memory locations;
14 if the resulting value is not within the values
15 in the limit memory locations, then performing an inverse
16 operation to restore the affected reservation register and
17 reporting a failure; and
18 if the resulting value is within the values in
19 the limit memory locations, then performing the operation
20 to update the actual value location, performing the
21 operation to update an unaffected reservation register, and
22 report a success.

1 14. A system for performing an operation within limits
2 upon a shared value stored in an actual value location
3 comprising:
4 a plurality of memory locations, two of said
5 memory locations being first and second reservation memory
6 locations, and two of said memory locations being limit
7 memory locations containing limit values.